Designation: E 2145 – 01

An American National Standard

Standard Practice for Modeling in Health Informatics[1]

This standard is issued under the fixed designation E 2145; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 Information models appropriately are used for analysis, design, and sharing a common understanding in health and healthcare information engineering, in healthcare process improvement, in building information systems, and in health informatics standards development.

1.2 The purpose of this practice is to identify best practices for the creation, use, and assessment of information models in the health and healthcare domain.

1.3 Included in this practice are recommended organizational policies and procedures, where modeling is best used in healthcare, and recommended modeling methods, best practices and evaluation criteria.

1.4 Excluded from this practice are detailed specifications of modeling techniques that are specified or described in other sources.

2. Referenced Documents

2.1 *ASTM Standards:*

E 1239 Guide for Description of Reservation/Registration-Admission, Discharge, Transfer (R-ADT) Systems for Automated Patient Care Information Systems[2]

E 1384 Guide for Content and Structure of the Electronic Health Record (EHR)[2]

E 1639 Guide for Functional Requirements of Clinical Laboratory Information Management Systems[2]

E 1715 Practice for Object-Oriented Model for Registration, Admitting, Discharge and Transfer (RADT) Functions in Computer-Based Patient Record Systems[2]

2.2 *ANSI Standards:*[3]

ANSI X3.172–1990 Dictionary for Information Systems

IEEE 1320.1-1998 Standard for Functional Modeling Language—Syntax and Semantics for IDEF0

IEEE 1320.2-1998 Standard for Functional Modeling Language—Syntax and Semantics for IDEF1X (Object 97)

2.3 *ISO Standards:*[3]

ISO 8601-88 Data Elements and Interchange Formats—Representation of Dates and Times

ISO/IEC 1087 Terminology Work—Vocabulary-Part 1: Theory and Application

ISO/IEC 11179 Information Technology—Specification and Standardization of Data Elements, Parts 1-6

ISO/IEC 2382 Information Processing Systems—Vocabulary

2.4 *FIPS Standards:*[4]

Information Modeling (IDEF1X) Federal Information Processing Standard 184. Gaithersburg, MD: National Institute of Standards and Technology. December 21, 1993. See also: IEEE 1320.2

Integration Definitions for Functional Modeling (IDEF0) Federal Information Processing Standard 183. Gaithersburg, MD: National Institute of Standards and Technology. December 21, 1993. See also: IEEE 1320.1

3. Terminology

3.1 The following sections present terms, definitions, and acronyms found in this practice and in various modeling activities.

3.2 *Definitions:*

3.2.1 *activity*, *n*—a group of logically related tasks performed for a purpose.

3.2.2 *alternate key attribute*, *n*—any candidate key of an entity other than the primary key.

3.2.3 *attribute*, *n*—a characteristic of an object or entity (see ISO/IEC 11179).

3.2.4 *attribute value*, *n*—a representation of an instance of an attribute (see ISO/IEC 11179).

3.2.5 *behavior column*, *n*—a column is a physical data model and relational database structure that is analogous to the attribute of a logical data model.

3.2.6 *concept*, *n*—a unit of thought constituted through abstraction on the basis of characteristics common to a set of objects, (see ISO/IEC 1087).

3.2.7 *concept activity model*, *n*—the activity component of the Concept Model.

3.2.8 *concept data model*, *n*—the data component of the Concept Model.

[1] This practice is under the jurisdiction of ASTM Committee E31 on Health Informatics and is the direct responsibility of Subcommittee E31.23 on Modeling for Health Informatics.

Current edition approved May 10, 2001. Published August 2001.

[2] *Annual Book of ASTM Standards*, Vol 14.01.

[3] Available from American National Standards Institute, 11 W. 42nd St., 13th Floor, New York, NY 10036. http://www.ansi.org.

[4] Available from National Technical Information Service (NTIS), U.S. Dept. of Commerce, 5285 Port Royal Road, Springfield, VA 22161. http://www.fedworld.gov.

3.2.9 *concept model*, *n*—also known as a conceptual schema from the ANSI/X3/SPARC Study Group for Database Management Systems.

3.2.10 *context*, *n*—a designation or description of the application environment or discipline in which a name is applied or from which it originates (see ISO/IEC 11179).

3.2.11 *data*, *n*—a representation of facts, concepts, or instructions in a formalized manner, suitable for communication, interpretation, or processing by humans or automatic means (see ISO/IEC 11179).

3.2.12 *data dictionary*, *n*—a database used for data that refers to the use and structure of other data; that is, a database for the storage of metadata (see ANSI X3, 172-1990).

3.2.13 *data element*, *n*—a unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes (see ISO/IEC 11179).

3.2.14 *data model*, *n*—a description of the organization of data in a manner that reflects an information structure (see ISO/IEC 11179).

3.2.15 *data steward*, *n*—a person or organization delegated the responsibility for managing a specific set of data resources, (see ISO/IEC 11179).

3.2.16 *dependent entity*, *n*—an entity that inherits one or more identifying attributes from another entity.

3.2.17 *domain*, *n*—the set of possible data values of an attribute (see ISO/IEC 2382).

3.2.18 *entity*, *n*—any concrete or abstract thin of interest, including associations among things (see ISO/IEC 2382).

3.2.19 *external schema*, *n*—an external schema is the representation of data at the system architecture presentation layer as viewed by the user when interacting or communicating with the information system.[5]

3.2.20 *foreign key attribute*, *n*—an attribute or combination of attributes of a child or category entity whose values match those in a primary key of a related or generic entity instance.

3.2.21 *function*, *n*—an activity, process or transformation identified by a verb that describes what would be accomplished.

3.2.22 *independent entity*, *n*—an entity that does not inherit identifying attributes from another entity.

3.2.23 *internal schema*, *n*—a schema of the ANSI X3/SPARC Three Schema[5] architecture in which views of the information are representations of data structure at the system architecture data layer.

3.2.24 *key attribute*, *n*—an attribute used to identify an instance of an entity.

3.2.25 *key inheritance*, *n*—the transmission of a key attribute from a parent or independent entity to a child or dependent entity.

3.2.26 *lexical*, *n*—pertaining to words or the vocabulary of a language as distinguished from its grammar and construction (see ISO/IEC 11179).

3.2.27 *logical data model*, *n*—a data model that presents a logical organization of data in the form of entities, attributes and relationships, especially where these components are normalized.

3.2.28 *metadata*, *n*—data that describes other data (see ISO/IEC 11179).

3.2.29 *non-key attribute*, *n*—an attribute that is not the primary or part of a composite primary key of an entity.

3.2.30 *object*, *n*—any part of a conceivable or perceivable world (see ISO 1087).

3.2.31 *object class*, *n*—a set of objects, ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behavior follow the same rules (see ISO/IEC 11179).

3.2.32 *object-oriented methodology*, *n*—any of several modeling and information management methods and techniques that employ objects rather than entities or tables.

3.2.33 *physical data model*, *n*—a data model that presents an organization of data in the form of tables, columns, and relationships.

3.2.34 *primary key attribute*, *n*—the candidate key selected as the unique identifier of an entity.

3.2.35 *property*, *n*—a peculiarity common to all members of an object class (see ISO/IEC 11179).

3.2.36 *relational data methodology*, *n*—any of several modeling and information management methods and techniques that employ structured relationships among entities or tables.

3.2.37 *row*, *n*—a row is a physical data model and relational database structure that contains an instance of data in a column.

3.2.38 *subject area*, *n*—a subject area is a portion of an entire data model which is created to facilitate understanding of a specific functional area or component task.

3.2.39 *table*, *n*—a physical data model and relational database structure that is analogous to the entity of a data model.

3.2.40 *transformational data model*, *n*—a data model that is optimized for performance on a specific technology.

3.2.41 *view (such as clinical)*, *n*—a collection of entities and assigned attributes (domain) assigned for some purpose.

3.3 *Acronyms:*

3.3.1 *ANSI*—American National Standards Institute.

3.3.2 *ASTM*—American Society for Testing and Materials.

3.3.3 *CODASYL*—Conference on Data Systems and Languages.

3.3.4 *ICOM*—Input, Control, Output and Mechanisms of IDEF activity modeling.

3.3.5 *ICT*—Information and Communications Technology.

3.3.6 *IDEF*—Integrated Definition Language.

3.3.7 *IDEF0*—The IDEF activity modeling language.

3.3.8 *IDEF1X*—The IDEF data modeling language.

3.3.9 *IDO*—Integrated Delivery Organization.

3.3.10 *IE*—Information Engineering diagramming methodologies.

3.3.11 *IEC*—International Electrotechnical Commission.

3.3.12 *ISO*—International Organization for Standardization.

3.3.13 *ISA*—Information Systems Architecture.

3.3.14 *UML*—Unified Modeling Language.

3.3.15 *DFD*—Data Flow Diagram.

3.3.16 *SML*—Standardized Modeling Language.

[5] ANSI Standards Planning and Requirements Committee-a layered model of database architecture comprising a physical schema, a conceptual schema, and user views.

3.3.17 *NIST*—National Institute of Standards and Technology.

3.3.18 *SQL*—Structured Query Language.

3.3.19 *CRC*—Class, Responsibility and Collaboration Approach.

3.3.20 *RDB*—Relational Database.

3.3.21 *RDBMS*—Relational Database Management System.

3.3.22 *CORBA*—Common Object Request Broker Architecture.

4. Summary of Practice

4.1 This practice describes policies and procedures, best practices for the creation, use, and assessment of health information models, and typical applications of information modeling in the health and healthcare domain.

4.2 The foundation for best practices in health information modeling may be derived from structure, process and outcome quality constructs **(1)**.[6]

4.2.1 *Quality Construct*—Information Modeling Area Supported.

4.2.2 *Structure*:

4.2.2.1 The organization component where modeling activities occur.

4.2.2.2 The composition of the modeling method or tool, its language and syntax.

4.2.3 *Process*:

4.2.3.1 Appropriate use of modeling.

4.2.3.2 Effective use of modeling methods.

4.2.3.3 Correct employment of the modeling technique.

4.2.4 *Outcome*—A quality health informatics model that provides a meaningful representation of past, present, or future reality.

4.3 The five dimensions **(2)** of model quality that derive from these quality constructs are as follows:

4.3.1 *Conceptual Correctness*—The model accurately reflects health or business concepts.

4.3.2 *Conceptual Completeness*—The model contains sufficient objects to describe the full scope of the target health or business domain.

4.3.3 *Syntactic Correctness*—The objects in the model do not violate syntactic rules of the modeling language.

4.3.4 *Syntactic Completeness*—All health or business concepts are captured at appropriate points in the modeling process.

4.3.5 *Enterprise Awareness*—The model depicts the entire enterprise or can be seamlessly integrated with other models across the entire organization.

4.4 *Enterprise Awareness*:

4.4.1 The foundation of a quality information model is the ability to describe the entire organization, either by itself or when combined with other comparable models.

4.4.2 Multiple models must seamlessly integrate across the entire organization and their descriptions must rationally flow from one model to another.

4.4.3 The Zachman Information Systems Architecture **(3)** is an effective tool to describe the totality of an organization from an informatics viewpoint. The Zachman ISA, therefore, is a suitable framework for identifying models that represent the informatics structure and processes of the organization.

4.5 *Conceptual Correctness*:

4.5.1 All models are representations of real objects or concepts.

4.5.2 The model must accurately describe the intended real organizational structures or processes, or both, and effectively communicate this description to anyone who understands the modeling method and syntax.

4.6 *Conceptual Completeness*:

4.6.1 Models describe the full scope or an element of the structures or processes, or both, of an organization or concept.

4.6.2 The model must contain sufficient objects to describe the full scope of the intended health or business domain.

4.7 *Syntactic Correctness*:

4.7.1 Each modeling method has a predefined language typically consisting of visual structures and symbols. Rules govern the manner in which these structures and symbols are assembled and interpreted.

4.7.2 The modeling method must use a recognized and preferably standardized set of rules **(4)**.

4.7.3 The model must adhere to the syntax of the modeling methodology.

4.8 *Syntactic Completeness*:

4.8.1 A modeling method may use different objects, symbols or structures at different times in the modeling process.

4.8.2 The model must employ the appropriate structures and symbols at the appropriate place in the model and at the appropriate time in the modeling process.

4.9 *Additional Technical Considerations*:

4.9.1 In addition to the dimensions of quality, several technical considerations impact the quality of an informatics model **(4)**.

4.9.1.1 *Nonredundancy*—A fact must be represented at only one point in the model.

4.9.1.2 *Business Rules Enforcement*—The model accurately reflects organizational, health, or business processes.

4.9.1.3 *Reuseability*—The content of the model is accessible to other than the originally intended users.

4.9.1.4 *Stability*—The model accepts changing health or business requirements without significant alteration in the structure of the model.

4.9.1.5 *Flexibility*—The model is capable of supporting change in health concepts or business processes.

4.9.1.6 *Simplicity*—The model is easily understandable and well presented.

4.9.1.7 *Implementable*—The model may be readily and economically used in analysis, design, development, or operation.

4.9.1.8 *Communicability*—The model readily conveys meaning to its users and readers.

4.10 *Assessing the Modeling Process and Model Quality*:

4.10.1 The dimensions of model quality that are derived from structure, process, and outcome quality constructs lead to best practices in the following areas:

[6] The boldface numbers in parentheses refer to the list of references at the end of this standard.

4.10.1.1 *Structure*—The organization desiring to employ modeling creates and maintains an environment conducive to the creation, evaluation, communication and use of quality information models.

4.10.1.2 *Process*—Modeling methods are appropriately selected and applied through effective use of modeling language and adherence to syntax to provide a meaningful, comprehensive, and balanced representation of real world health or business concepts.

4.10.1.3 *Outcome*—The resulting model(s) exhibit a high degree of excellence and wholeness are identified by author, participants, technique and approach, perspective, view and subject areas, and may include test data for evaluation purposes.

4.10.2 Annex A1 of this practice provides checklists for assessing the quality of models and modeling activities in terms of structure, process, and outcome.

5. Significance and Use

5.1 Modeling is increasingly used in the healthcare industry to develop a common understanding of processes, functions, activities, and supporting data. Typical users of such models are healthcare systems developers, health, and healthcare operations researchers and analysts, educators, and executives.

5.2 Information models are regarded widely as beneficial by saving the cost through risk reduction and elimination of redundancy. Information models convey ideas and facilitate analysis and understanding of complex processes and structures. These models form the basis for software engineering practices that build databases, redefine organizational processes, and develop standards.

5.3 This practice provides a practical means for developers and users of health information models to employ appropriate modeling methods and to objectively determine model quality.

5.4 *Background*:

5.4.1 Models are representations of past, existing, or contemplated reality. Models may assist in the explanation or analysis of complex structures and processes that may exceed human capacity for direct visualization or understanding. Models enable a focus on the key elements of a process or structure while ignoring confounding or irrelevant elements. As such, models make an explicit statement of the meaning of the reality being modeled.

5.4.2 Integrated information engineering models provide a coherent view of the activities and data of an organization or enterprise **(5)**. Activity models identify the fundamental tasks performed in a function. Process models accurately describe the detailed collection of activities within an organization. Data models are derived from and support the functions described in activity models. Object models collectively characterize the processes and data required to understand the operation of an organization and to construct information systems that support that operation. Models may be textual, graphic, or mixed graphic and text forms, including, flowcharts, state diagrams, data flow diagrams, entity-relationship diagrams, and related techniques develop an understanding of business processes and the transformation of data through these processes **(6)**.

5.5 *Frameworks for Relating Models to Systems*:

5.5.1 Since proposed by Zachman in 1987 **(3)**, the Information System Architecture (ISA) has become an accepted means to view and to blueprint information systems within an entire organization, designing business practices, and changing enterprise strategies. Zachman developed this ISA as a "framework" to describe and interrelate the components of an information system to the entire organization. The Zachman ISA has been adapted for the Healthcare sector: Table 1 presents the ISA as a framework of healthcare related informatics standards that may apply in a given healthcare enterprise.

5.6 *Principles and Approaches to Modeling*:

5.6.1 Modeling activities have been prevalent throughout the history of the engineering disciplines. This rich history indicates four basic principles **(7)**, as:

5.6.1.1 The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped.

5.6.1.2 Every model may be expressed at different levels of precision.

5.6.1.3 The best models are connected to reality.

5.6.1.4 No single model is sufficient; every nontrivial system is best approached through a small set of nearly independent models.

5.6.2 The three approaches to information modeling are characterized as top-down, bottom-up, and inside-out **(8)**.

5.6.2.1 The "top-down" approach follows a progression of activities that first analyzes and understands functionality of the domain-of-interest and only then progresses to model preparation. The top-down paradigm typically consists of preparing a high level conceptual model, consisting of activity model and an entity-relationship data model. Logical data models may be constructed as appropriate for the standard and healthcare domain. The strength of this approach is it provides a broadly-based and comprehensive set of rationally derived models that are generally appropriate across the spectrum of healthcare activities. This paradigm saves time and resources in development.

5.6.2.2 The "bottom-up" approach reflects the experience of subject matter experts in a variety of specialized subdomains. These panels of experts prepare small models that are subsequently integrated. The strength of this approach is that it addresses detail that can be easily converted to a specific application, and may be tuned or optimized for a segment of the healthcare environment. Often, it is used when the entire domain is not well understood or in the absence of a business model foundation. The bottom-up paradigm produces narrowly focused model components requiring extensive effort for component integration.

6. Determining the Need for Information Modeling

6.1 In health informatics, these principles may be applied to modeling for content, modeling in information systems development, modeling in standards development.

6.1.1 *When and Where to Use Modeling Techniques for Content*—The complexity and extensiveness of current health information and health care information systems preclude a comprehensive understanding to the depth of detail frequently

TABLE 1 The Zachman Information Systems Architecture

	Why	When	Who	What	How	Where
			Vision [Guidelines]			
Scope (contextual)	Goals (motivation) 1. Personal/Public Healthcare delivery business case	Events (time) 2. Identification of significant care/care delivery events	Stakeholders (People) 3. Essential health service organizations and functions	Values (content) 4. Description of important healthcare service and care delivery information	Processes (function) 5. Important healthcare and care delivery services	Locations (Network) 6. Identification and description of organization and individual locations
			Design [Standards]			
Enterprise model (Conceptual)	Objectives	Timeline	Organization	E-R Data Model	Process Model	Interface Architecture
	7. Personal health benefit and care delivery business objectives	8. Sequence and timelines of healthcare services	9. Healthcare information workflow	10. Semantic description of healthcare processes	11. Conceptual activity model of healthcare delivery	12. Structure and interrelationship of healthcare facilities
System model (Logical design)	Requirements	Phases	Hierarchies	Logical Data Model	Data Flow	Network Model
	13. System Functional Requirements	14. Healthcare event phases and process components	15. Healthcare information system human-system interface architecture	16. Logical data model for healthcare information	17. Application architecture with function and user views	18. Connectivity and distributed system architecture
			Implementation [Standards]			
Technology model (Physical design)	Knowledge Design	Control Structure	Human-Technology Architecture	Physical Data Model	Structure Chart	System Architecture
	19. System Operational Requirements	20. Healthcare information system control structures	21. Healthcare information system human system interface description	22. Physical data model for healthcare information	23. System Design, language specification and structure charts	24. Health system information network detailed architecture
Components (Modules and subsystems)	Knowledge Definition	Timing Definition	Security Architecture	Data Dictionary	Program Description	Network Architecture
	25. Technical Requirements	26. Healthcare Information System component timing descriptions	27. System Security Architecture and Operations	28. Healthcare Information Metadata and DBMS scripts	29. Code Statements, Control blocks, DBMS stored procedures	30. Physical data network components, addresses and communication protocols
			Operation [Standards]			
Functioning system	Strategy	Schedule	Organization	Data	Function	Network
	31. Technology Operational Requirements	32. Healthcare information system operation schedules	33. IS participant description	34. Functioning database, knowledgebase	35. User procedural system and documentation	36. Operating health system communication network

required. Formal methods are needed to facilitate the development and sharing of a common understanding about such information and systems. The following sections suggest where and how a given modeling method or technique may be appropriately used to model content, to develop systems, and to develop standards. These sections also outline how to recognize where and how to select a particular modeling technique for application to a specific problem and enterprise, and how to document the representation health information and healthcare systems concepts.

6.1.2 *Integrated Delivery Organizations (IDO)*—Almost all Integrated Delivery Organizations (IDOs) will have facilities and settings of care that range across the familiar resident (inpatient) and nonresident (ambulatory) facilities and settings. Historically, these have been treated differently, but in Guide E 1239, the concept of a common structure for the care record applicable to all settings of care is introduced and is carried forward into Guide E 1384. That concept is applicable to all of the information subdomains and healthcare professional specialty views of the underlying data structures and representations. In order to achieve composable component models from which to build an entire enterprise model, common underlying concepts must be consistently used in each subdomain of an integrated composed domain information model. These common concepts applied to the familiar ancillary services and activities are dealt with below. The application of modeling techniques to an IDO should only proceed as part of a defined project within the Life Cycle Processes for a system if the range of participants needed to model an organization of this scale are to be systematically engaged in a manner that will lead to a useful outcome.

6.1.3 *Resident Healthcare Facilities*—The most familiar resident facility is the hospital, which has a number of specialty configurations, many of which are described in Guide E 1384 and each configuration has different patterns of specialty participation. Using the methods, techniques and tools in Sections 6 and 7, the facility must be dissected into its constituent specialty views and the information subdomains identified.

6.1.4 *Population/Public Health*—The ability to assess population and public health care is critical to any health care information system. Modeling of these components helps ensure the development of an inclusive health Domain Information Model. In addition, population and public health care needs may have significant architectural impact on the information system. Modeling of these partitions will identify those essential information subdomain components that must be addressed through the development process.

6.1.5 *Ambulatory Care Facilities*—Ambulatory care facilities serve the widest variety of settings and enterprise arrangements, many of which require interoperation of the facility informatics domain with inpatient facilities. Understanding the information architecture which best serves each practice setting requires careful modeling in order to understand the information, the data structures, the data representations, the information services and system responsibilities and constraints. Part of this architecture can be built upon common frame works that are documented in the healthcare informatics standards, such as ADA 1000 Series, Guide E 1384, Practice E 1715, Health Level 7, and used as a core for a set of enterprise models that document how the individual enterprise operates and interacts with the external world. Careful consideration of the Zachman ISA Framework (see 4.5) helps document the modeling steps that may be needed.

6.1.6 *Home Healthcare Facilities*—In the emerging healthcare arrangements, care at home by visiting practitioners is increasing. The information that documents care by family members or other residents in a patient's home, in addition to that rendered by visiting practitioners will be part of the health Domain Information Model that must be represented. Both content concepts and implementation technology must be represented for this setting of care so that both clinical care and resource management concepts are included.

6.1.7 *Laboratory Components*—Modeling of not only the information architecture internal to the clinical laboratory of the but also the other enterprise environments it serves is also appropriate. Guide E 1639 deals with the clinical laboratory enterprise environment and the internal requirements of the of the clinical laboratory information domain. This standard depicts the modeling methods that help the clinical laboratorian understand the internal structures and activities and how to relates these to the practitioner customers that they serve. Guide E 1239 and Practice E 1715 depict the key foundation data structures for registration admitting, discharge functions that are common to both the clinical laboratory and to the EHR information domain of the practitioner.

6.1.8 *Pharmacy Components*—Pharmaceutical care, in all settings, is becoming recognized as a contributing information subdomain that is a component of clinical views of the patient care record. In addition to the attributes recorded for the care record, additional resource management attributes within the pharmacy itself and its relationship with the financing subdomain, must be modeled accurately in order to reflect the obligations, responsibilities and constraints of the information services involving this pharmaceutical care Subdomain.

6.2 *When and Where to Use Models for System Development*:

6.2.1 At a high level, all development methodologies employed in software engineering share a sequence of four essential activities from project inception to implementation **(9)** as illustrated in Fig. 1.

6.2.2 *Why Use Modeling in System Development*:

6.2.2.1 Software engineering is heavily dependent upon modeling activities.

6.2.2.2 System development models may consist of elementary structures like an organizational chart to detailed structural representation of a software package. Models typically are employed in the planning and analysis phase of software development, in system and software design, and in the coding process.

6.2.2.3 Booch, et. al. **(7)**, state that the single fundamental reason for modeling a system is to better understand the system being developed, achieving four aims:

(a) To visualize an existing or desired system;

(b) To specify the structure and behavior of a system;

(c) To provide a guiding template for system construction; and,

(d) To document the decisions made in system development.

6.2.2.4 Pressman notes three additional reasons for modeling **(10)** in system development:

(a) The business problem and solution can be approached incrementally, thus reducing the probability that a major feature will be misinterpreted or overlooked;

(b) The customer or user can more easily review the analysis and design, to help preclude or eliminate areas of ambiguity and misunderstanding; and,

(c) The development proceeds in a logical, efficient and more readily manageable manner through the migration of models from analysis through design and development.

6.2.2.5 Most importantly, models greatly enhance quality in system development. Models enable the developer to proactively build-in quality during design rather than reactively assess quality during development. As models are transformed from analysis through physical software, sets of quality criteria can be created for each modeling step, with the models reviewed against these criteria incrementally. Errors in requirements definition, design, and development can be uncovered easily and corrected at the earliest possible time with less effort and delay than if corrected during module or integration testing.

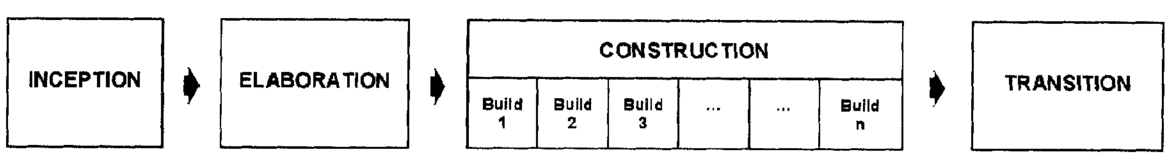

FIG. 1 High Level View of Software Development Steps

6.2.3 *Modeling in Systems Inception*—Analytical models are essential to the systems inception and planning. In the inception phase, models help develop the business rationale and communicate this rationale in the system business case. Semantic and diagrammatic models help explain the thought processes leading to a system proposal. Activity models are useful to design future business activities and business process improvement. Financial models and simulations are important components of the business case and risk prediction **(11)**, and for software engineering project management **(12, 13)**, and estimation **(14)**.

6.2.4 *Modeling in Systems Elaboration*—The elaboration phase of the systems development paradigm includes analysis and high level design. Models are essential to system elaboration by analyzing, recording, and detailing business process and system functionality. Typical analytical and high level design models include the following:

6.2.4.1 *Semantic Model*—a verbal depiction of processes or features, frequently as a set of functional requirements or a functional description.

6.2.4.2 *Flowchart*—A diagrammatic description of related activities and system components.

6.2.4.3 *Activity Model*—A graphic presentation of the tasks performed in existing and/or desired systems.

6.2.4.4 *Data Flow Diagram*—A graphic representation of processes, data stores, and the sources and destinations of data.

6.2.4.5 Conventional flowcharting proved a valuable modeling tool for analysis and design of systems employing procedural languages. Although largely supplanted in the client-server environment by more capable methods, conventional flowcharting continues to have a role in requirements analysis. Activity modeling describes the high level activities through detailed tasks and may include features, such as activity-based costing.

6.2.5 *Migration of Models Through System Construction*—A software engineering methodology provides development paradigms as a procedural framework for the design, construction, testing and implementation of systems. Within a development methodology, modeling methods prescribe the symbology and heuristics used to create models of varying levels of detail. Automation, via computer-assisted software engineering (CASE) tools, facilitates the modeling process and provides the vehicle that achieves transformability. The transformation from analysis to a functioning system is termed forward engineering while transformation from a functioning system to analytical or design models is termed reverse engineering.

6.2.5.1 The benefit of using standardized modeling methods is the software models that are created are easily understandable and readily transformable to a functioning system **(10)**. Migration of Models facilitates the understanding of a health or healthcare concept and the implementation of information applications, databases or systems. As illustrated in Fig. 2, multiple models follow a logical progression from Conceptual to Logical to Physical **(8)**.

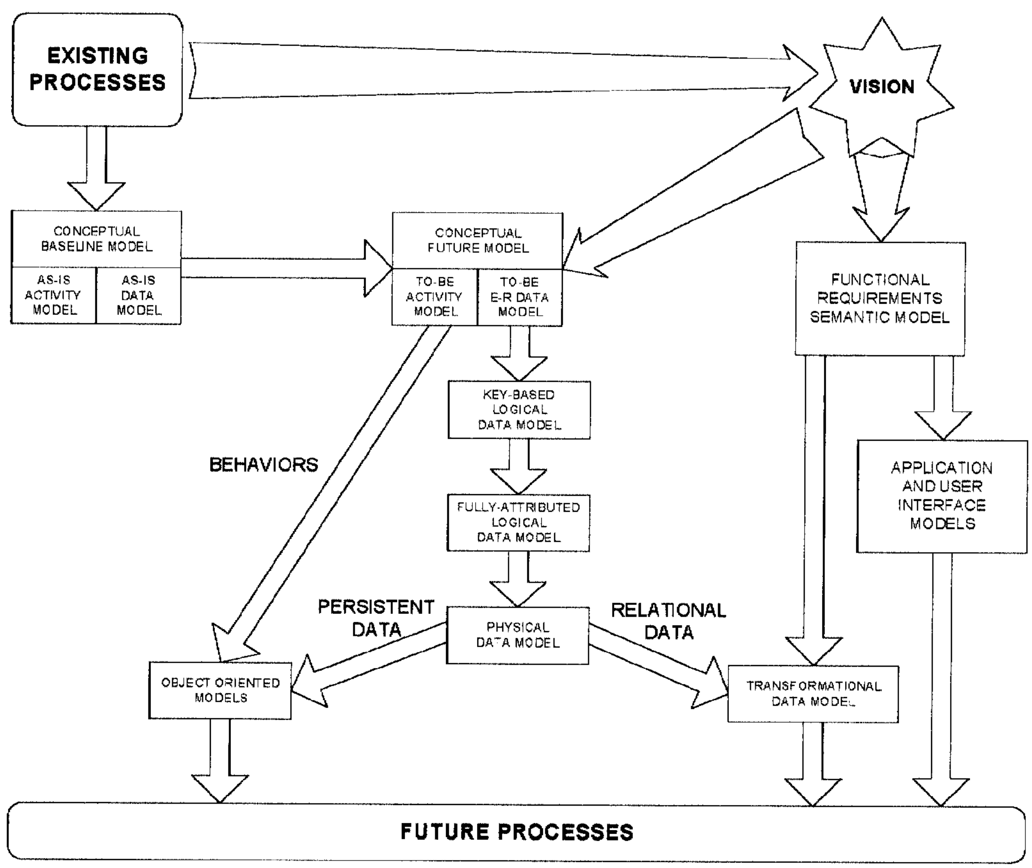

FIG. 2 Migration of Models in System Development

6.2.5.2 The systems development process originates with either an existing system in need of improvement or a vision of a new or reengineered system. The chain of models begins with conceptual modeling where activities and data are modeled at a high level of detail.

6.2.5.3 Conceptual models present the high level view of an activity and its supporting data. Conceptual models document the existing system or conditions as a Conceptual Baseline Model or a desired system or condition as a Conceptual Future Model. These conceptual models consist of an activity model and a data model.

6.2.5.4 The Conceptual Activity Model is a high level representation of the fundamental business activities performed by an organization. The Conceptual Activity Model describes the job design, task mix, and workflow in the business architecture **(15)** of a healthcare organization. The Conceptual Activity Model can represent the business architecture from a current ("As-Is"), or possible future ("To-Be") perspective. The Conceptual Activity Model may take a linear form, such as in the IDEF0 method, a closed loop as in business cycles, or a linear-cyclic mix as in conventional flowcharting. For improved understanding, one or more Conceptual Activity Models may be constructed from different perspectives or using differing methodologies, or both.

6.2.5.5 The Conceptual Activity Model may progress through a series of decompositions where model component activities are described with increasing detail. A detailed activity model may be used for a variety of purposes, such as:

(a) To describe the task detail of a particular job;

(b) To elucidate system functional requirements and object behaviors; or,

(c) To link economic aspects to the task detail as in activity-based costing.

6.2.5.6 The Conceptual Data Model is a high level representation of the essential data required to enable operation of the business function. Most typically the Conceptual Data Model takes the form of an Entity-Relationship Diagram where the entities identify the fundamental things required to perform a business function and the relationships denote the business rules or reasons these entities operate with each other. An "As-Is" conceptual data model describes the fundamental data used in the existing system while a "To-Be" conceptual data model describes the data anticipated for a future system.

6.2.5.7 As with decomposition of the activity models into increasing detail, data models progress from a conceptual level to a detailed representation of data as would be used in a functioning database. The Conceptual Data Model is an entity-relationship diagram that contains numerous nonspecific, many-to-many relationships. Normalization transforms the Conceptual Data Model through removal of nonspecific relationships and, along with adding candidate keys to independent entities, is the first step in preparing a Logical Data Model **(8)**. Ensuring that these keys are correct and migrate appropriately through the model are critical to developing referential integrity in the logical data model and physical database.

6.2.5.8 Adding candidate nonkey data attributes to appropriate entities the logical data model creates a fully attributed logical data model. All nonkey attributes are reviewed to ensure that all required data is represented. The structure of the fully attributed logical data model is then examined and modified as needed to ensure that a Third Normal Form has been achieved. At this point the logical data model should be reviewed for quality aspects to ensure that:

(a) All entities and attributes are defined;

(b) Relationships are stated and defined;

(c) Cardinality is appropriately stated; and,

(d) Referential integrity is maintained throughout the model.

6.2.5.9 While the conceptual and logical data models are technology independent, the physical data model is expressed in terms of a specific technology. The logical data model is transformed into a physical data model by incorporating data characteristics typically found in a data dictionary and in the form of the target relational database management system, object-relational data management system, or object oriented data management system.

6.2.5.10 Before a physical data model becomes a functional relational database it typically undergoes a transformation to optimize the performance of the data structure in the anticipated operational environment. At this point a Transformational Data Model **(8)** is created where the data structure is heavily denormalized. The Database Management System model, the actual database, is built from this transformational model.

6.2.6 *Organizational Issues*—The structure and process elements of the model quality paradigm involve the creation and maintenance of an environment and procedures that facilitates creation, evaluation, communication, and utilization of information models. For the system development organization there should be a component responsible for the creation and maintenance of models. The organization should establish policies for modeling and standardize the model development process, methodologies, and tool sets. These policies constitute internal, or organizational standards or conventions, consisting of the following:

6.2.6.1 *Syntactic Conventions*—The library of model symbols to be employed;

6.2.6.2 *Positional Conventions*—The manner in which symbols are displayed or presented in the model; and,

6.2.6.3 *Semantic Conventions*—The grouping of model components according to meaning.

6.2.6.4 The organization should provide a battery of standardized modeling methods from which the software developers can select as appropriate to the needs of a specific development project.

6.3 *When and Where to Use Models for Standards Development*:

6.3.1 Organizations may develop their own internal standards, such as standard operating procedures, or they may develop standards for more general use in the health care industry. Modeling facilitates the development, communication and understanding of both internal and industry-wide standards. Developers of internal and external standards may use models to prepare and present a common reference to their work. Furthermore, consumers may find models facilitate the implementation and use of standards by their organizations by providing standards users a blueprint from which they can tailor and optimize model components for implementation.

6.3.2 Top-down or bottom-up approaches may be used to develop both internal and external standards based on common models of shared activity, process and data, and an object-oriented approach may be used to develop internal standards. The middle-out approach is only suitable for developing internal or organizational standards.

6.3.3 *The Top-Down Approach for Standards Modeling*:

6.3.3.1 The top-down approach is required for preparing external standards that address the entire healthcare domain, for example, electronic patient health information, or large portions of this domain.

6.3.3.2 In the top-down approach a working group assembles to begin modeling by analysis of a single, overarching business or health domain feature or principle. When dealing with healthcare delivery, finance or data models for example, this feature may be the nature and value of individual health. Such overarching features are difficult and time consuming to appropriately develop yet these features and principles are essential to the creation of a comprehensive, rational model.

6.3.3.3 A top-down approach also can be used for modeling domain components. In this approach a working group of subject matter experts adds precise and accurate detail to those relevant portions of the general model that have been filtered into the subject area. This approach provides efficient model development and ease of integration into the larger general domain model.

6.3.4 *The Bottom-Up Approach for Standards Modeling*:

6.3.4.1 The bottom-up approach is appropriate for preparing external standards that cover a small portion of the health care domain, for example, telepharmacy data transactions. The bottom-up approach also is suitable for preparing internal, local, or organizational standards of any scope.

6.3.4.2 In the bottom-up approach, draft models are developed by working groups representing the interests of domain components, for example, ambulatory care. The bottom-up approach produces detailed models of that domain component usually very rapidly since the working group members can focus on a very small segment of the healthcare domain. This approach can develop standards. components requiring extensive effort for integration into a larger, more comprehensive model. This component integration is very time consuming, runs greater risk of error than the top-down approach, and perpetuates gaps in the model where domain interests have been omitted. The delays in filling in domain gaps and integration pose a substantial risk that on approaching completion, a comprehensive, bottom-up model of the entire domain will not accurately represent contemporary reality.

6.3.5 *Object-Oriented Modeling in Standards Development*:

6.3.5.1 An object model may be useful to prepare local and internal standards for use by an organization or group of organizations sharing nearly identical business processes. The strength of the object model derives from its description of data in the context of processes specific to the organization being modeled. The object model allows description or development of a functioning system with minimal effort to optimize or tune the application or data architecture.

6.3.5.2 The object-oriented approach is ideal for consortia with narrowly focused domains where the object model will highlight opportunities for interoperability and component reuse. The commonality that underpins the object oriented approach rapidly fades with divergence of these shared business processes. Since healthcare business and care delivery processes are geographically bounded and tailored by organizations to meet specific local needs, object models serve best as local representations of that shared reality.

6.3.6 *Standards Development Organizational Issues*—As with systems development, SDOs preparing information standards need an organizational component responsible for the creation and maintenance of a modeling environment. This organizational component establishes modeling internal standards for the organization, selects modeling methodologies and tools, and assists working groups in their development of models.

7. Best Practices for Health Informatics Models

7.1 *Required Foundation Features*—Every model must have a set of descriptive foundation features to aid in the communication and understanding of the model.

7.1.1 *Model Identification*—Every model must be identified by the following parameters:

7.1.1.1 Name or title of the model;

7.1.1.2 Version number of the model;

7.1.1.3 Name of the person who created or edited the model; and,

7.1.1.4 Date on which the model was created or edited.

7.1.2 Parameters required by modeling standard **(16)** should be used; otherwise present the model parameters as shown in Table 2.

7.1.3 Additional optional information, such as sponsoring organization, may be included. The format for the date entry should be ISO 8601 compliant **(17)**. Some modeling methods and CASE tools, such as for the IDEF1X data modeling method, present a standardized "kit" that includes these identification parameters.

7.1.4 *Modeling Technique*—Model builders must state the modeling technique used to create the model, for example, IDEF1X data modeling.

7.1.5 *Approach*—Model builders must include the approach (as top-down or bottom-up) for reference.

7.1.6 *Views and Perspectives*—Model builders must include the viewpoint for reference. For example, a data model constructed from the viewpoint of a system to be used by a practitioner will appear differently than a data model of the same system when used by a business manager.

7.1.7 *Clarity of Presentation*—Models must be understandable and convenient to use. Objects sized and arranged on the screen or page to be easily read at normal reading distance.

TABLE 2 Example Model Identification Table

Identification Parameter	Example Identification Entry
Model name	Hospital Formulary Logical Data Model
Version	2.3
Author/editor	Anna Phalaxis
Date created/edited	1999 Dec 20

Normally no more than 20 objects should be displayed on a printed 8.5 by 11-in. or A4 page. Where the model is too large, or too complex, or both, for clear presentation, it should be decomposed into multiple submodels or subject areas as logical groupings of diagram objects, often to fit a specific business function.

7.1.8 *Including Sample Data for Quality Assessment*—Models, such as financial and data models, may be bench-tested through a process called instantiation. Instantiation involves insertion of real world values into the model elements for explanation and testing purposes. For example, while an integral component of a Physical Data Model, metadata may be included in the Logical Data Model to facilitate testing and validation. Likewise, test financial data may be included in an activity model to validate activity-based costing features.

7.2 *Health Informatics Structural Models*:

7.2.1 Structural models are frequently used to represent, analyze and communicate physical features of a health care organization. Structural models may include the following:

7.2.1.1 Organization charts;

7.2.1.2 Network diagrams;

7.2.1.3 Facility blueprints; and,

7.2.1.4 Instrument schematics.

7.2.2 The collection of structural models forms the core of the health care organization control architecture. Quality dimensions for healthcare structural models are outlined in Table 3. Required syntactic correctness practices are that the modeling process and model fully conform to consensus or industry standard practices for the methodology or technique employed to prepare structural models.

7.3 *Health Informatics Function Models*:

7.3.1 The function-driven approach to modeling has a basis in the activities performed by an organization **(18)**. These function models reflect the activities and tasks performed at all levels of an organization. Function models may represent the activities alone, activities combined with data or activities with measures, such as cost breakouts **(19)**.

7.3.2 The most basic function model is the semantic model, a narrative representation of activities or tasks performed in an organization. The semantic model is used frequently as a system functional requirements listing. Graphical function models **(6)** include the following:

7.3.2.1 Conventional flowcharting;

7.3.2.2 Business cycle diagramming;

7.3.2.3 Value chain;

7.3.2.4 Data Flow Diagramming; and,

7.3.2.5 IDEF0 Activity Modeling.

7.3.3 These modeling methods are well described in the literature **(6, 20)**.

7.3.4 Quality dimensions for healthcare functional models are outlined in Table 4.

7.3.5 Required syntactic correctness practices for functional models are as follows:

7.3.5.1 Objects displayed according to specific methodologies, for example, sharp-cornered boxes for IDEF0 activities.

7.3.5.2 Object names displayed in upper case within the object.

7.3.5.3 Connector names, such as IDEF0 ICOMs and DFD data flows, displayed in mixed case.

7.3.5.4 Connector flow direction is denoted by an arrow at the destination end of the connector line.

7.3.6 Object numbering, such as for Data Flow Diagram and IDEF0 objects, is optional.

7.4 *Health Informatics Data Models*:

7.4.1 The data-driven approach to modeling also is known as the Information Engineering Approach **(5)**. Data schema diagrams were presented by Bachman in the 1960s, and subsequently improved, as rectangles for record types and connecting arrows for relationships **(21, 22)**. Following this work, the Entity-Relationship approach was described by Chen as rectangles for entities and diamonds overlaying connecting arrows to detail the type of relationship **(23)**. Data models in particular have become widely recognized as essential to design quality databases **(24, 25)**. Some authors have compiled sets of models in other industries that they present as guidelines or references for reusable data structures **(26)**. Data Modeling techniques using the IE, Chen, and IDEF1X notations have been well described in the literature **(6, 20)**

7.4.2 Quality dimensions for healthcare data models are outlined in Table 5.

7.4.3 Syntactic correctness practices for data models are described as follows:

7.4.3.1 Independent entities displayed as boxes with sharp corners.

7.4.3.2 Dependent entities displayed as boxes with rounded corners.

7.4.3.3 Entity names are in upper case on the top left edge of the entity box.

7.4.3.4 Attribute names are in mixed case.

TABLE 3 Quality Dimensions and Examples for Health Informatics Structural Models

Structural Model Quality Dimension	Model Quality Characteristics	Model Quality Dimension Example
Enterprise awareness	The structural model depicts the entire enterprise or object, or a component that can be seamlessly integrated with other models across the entire organization or component.	A blueprint of a hospital or an surgical suite as one of a complete set of blueprints of the hospital.
Conceptual correctness	The structural model accurately reflects health or business concepts.	A hospital department organization chart is maintained current to accurately reflect changes in staffing and assignments.
Conceptual completeness	The structural model contains sufficient objects to describe the full scope of the target health or business domain.	An electrical schematic of a defibrillator displays all components
Syntactic correctness	The objects displayed in the structural model do not violate syntactic rules of the modeling language.	A local area network diagram for a managed care organization uses industry standard conventions to depict network components.
Syntactic completeness	All health or business structural concepts are captured at appropriate points in the modeling process.	A provider network organizational diagram is prepared following determination of the operating relationships among the providers.

TABLE 4 Quality Dimensions and Examples for Health Informatics Functional Models

Functional Model Quality Dimension	Model Quality Characteristics	Model Quality Dimension Example
Enterprise awareness	The functional model depicts the entire enterprise or object, or a component that can be seamlessly integrated with other models across the entire organization or component.	A comprehensive IDEF0 activity model is prepared for an entire managed care organization.
Conceptual correctness	The functional model accurately reflects health or business concepts.	A business cycle model depicts the activities and value-added in an outpatient care delivery setting.
Conceptual completeness	The functional model contains sufficient objects to describe the full scope of the target health or business domain.	A data flow diagram for a hospital pharmacy service presents all information sources and destinations, data stores and processes.
Syntactic correctness	The objects displayed in the functional model do not violate syntactic rules of the modeling language.	Components of an IDEF0 activity model for a clinical laboratory conforms to FIPS Standard 183.
Syntactic completeness	All health or business functional concepts are captured at appropriate points in the modeling process.	A semantic model for an outpatient surgical service management system is prepared as a list of functional requirements by interviewing key participants.

TABLE 5 Quality Dimensions and Examples for Health Informatics Data Models

Data Model Quality Dimension	Model Quality Characteristics	Model Quality Dimension Example
Enterprise awareness	The data model depicts the entire enterprise or object, or a component that can be seamlessly integrated with other models across the entire organization or component.	A Chen diagram includes entities and attributes that support the entire operation of an IPO.
Conceptual correctness	The data model accurately reflects health or business concepts.	A conceptual data model for a primary health clinic contains entities, attributes, and relationships that accurately describe business and clinical practices of the clinic.
Conceptual completeness	The data model contains sufficient objects to describe the full scope of the target health or business domain.	A fully attributed IE logical data model presents all entities and attributes required for a materials management data system.
Syntactic correctness	The objects displayed in the data model do not violate syntactic rules of the modeling language.	An IDEF1X logical data model for a dental clinic conforms to FIPS 182.
Syntactic completeness	All health or business data concepts are captured at appropriate points in the modeling process.	A series of Joint Application Development session were held to sequentially prepare a conceptual data model, a key-based logical data model and a fully attributed logical data model for a clinical pathology service.

7.4.3.5 Identifying key attribute names are located above a horizontal line inside the entity box.

7.4.3.6 Nonidentifying key attribute names located below a horizontal line inside the entity box and above nonkey attribute names.

7.4.3.7 Nonkey attribute names located below a horizontal line inside the entity box and below the nonidentifying key attribute names.

7.4.3.8 Identifying relationships presented as a solid line.

7.4.3.9 Nonidentifying relationships presented as a broken line.

7.4.3.10 Relationship cardinality displayed according to methodology, for example, a dot with a character symbol for IDEF1X, a "crowsfoot" for IE notation, etc.

7.4.3.11 Relationship names are optional; when displayed relationship names are in lower case.

7.4.4 Optional comment fields may be included in data model diagrams. If included comments should be displayed in mixed case complete sentences within a rectangular box.

7.4.5 *Conceptual Data Modeling*—Quality dimensions for a Conceptual Data Model include the following:

7.4.5.1 Conceptual completeness requires all essential entities be represented and all relationships among these entities be correct representations of the functional interrelationships of the business objects represented in the model.

7.4.5.2 Conceptual correctness requires all essential entities be derived from and support a prerequisite conceptual activity model.

7.4.5.3 Syntactic completeness requires all entities and attributes are defined or described for clarity of understanding.

7.4.5.4 Syntactic correctness requires that there is strict adherence to the modeling methodology requirements.

7.4.6 *Logical Data Modeling*—A logical data model flushes out the detail of a conceptual data model without consideration of implementation technology. Quality dimensions for key-based Logical Data Models include the following:

7.4.6.1 Conceptual completeness requires all entities reflect or expand upon those in the precursor conceptual data model.

7.4.6.2 Conceptual correctness requires candidate entity keys are appropriate, descriptive, and unique.

7.4.6.3 Syntactic completeness requires that all many-to-many relationships are converted through interposition of associative entities, and all entities, key attributes, and relationships are defined or described.

7.4.6.4 Syntactic correctness requires all entities and relationships are normalized through the Third Normal Form; referential integrity is maintained through proper migration of all key attributes throughout the model; and the model presentation strictly adheres to the modeling methodology employed.

7.4.6.5 Quality dimensions for Fully Attributed Logical Data Models include the following:

(a) Conceptual completeness requires that the fully-attributed model provides detail that expands upon the precursor key-based logical data model; all reference entities contributing identifying and nonidentifying foreign keys are displayed; and all required descriptive detail is contained in nonkey attributes.

(b) Conceptual correctness requires that all nonkey attributes contribute detail to the appropriate entities, and that entity subtypes appropriately reflect choices in populating data structures.

(c) Syntactic completeness requires that all reference entities are named and defined; all reference entity foreign key and nonkey attributes are named and defined; and all subtype delimiters, entities, and attributes are named and defined.

(d) Syntactic correctness requires normalization through the Third Normal Form is preserved; referential integrity is maintained through proper migration of all identifying and nonidentifying key attributes throughout the model; all entities, key attributes, and relationships are defined or described; and the model presentation strictly adheres to the modeling methodology employed.

7.4.6.6 *Physical Data Modeling*—Physical data models expand upon the fully attributed logical data model to provide detail sufficient to implement a relational database or construct objects. Quality dimensions for Physical Data Models include the following:

(a) Conceptual completeness requires preservation of the data structures developed in the precursor logical data model, and addition of metadata appropriate for the target database management system.

(b) Conceptual correctness requires metadata such as field length be appropriately specified for the intended function of the attribute.

(c) Syntactic completeness requires all metadata be included in the model.

(d) Syntactic correctness requires specification of metadata consistent with the target database management system and modeling methodology.

7.5 Health Informatics Process and Object Models comment on object modeling is new, volatile, and expect evolution. Quality dimensions for healthcare object models are outlined in Table 6. Syntactic correctness practices for object models are described as follows:

7.5.1 *Process Characterization*—A process is a set of work tasks that accomplish a purpose. The process model depicts these tasks as workflow along with any required inputs and outputs. Workflow, illustrating the sequence in which these tasks are performed, often involves recursion where a specific task or activity may be performed more than once and following more than one preceding activity. Quality dimensions for Process Models include the following:

7.5.1.1 Conceptual completeness requires identification of the component tasks or activities; representation of the sequence in which tasks are preformed; and, identification of any objects that are the inputs or outputs of these tasks.

7.5.1.2 Conceptual correctness requires that the component tasks accurately reflect the workflow being represented.

7.5.1.3 Syntactic completeness requires that the process inputs and outputs are named and defined; all component tasks or activities are named and defined; and, all process flows among tasks or activities are named or defined.

7.5.1.4 Syntactic correctness requires that tasks or activities are represented as sharp-cornered boxes; inputs and outputs to and from these tasks are represented as arrows connecting the task boxes; input and output arrows indicate the sequence in which tasks are performed; input and output arrows are named and defined; and, the model presentation strictly adheres to an additional requirements of the specific modeling methodology employed.

7.5.2 *UML Modeling*—One notation, originated by Jacobsen, used for developing scenarios employed in analysis of a problem is "use cases ()". It depicts the "actors" or individuals involved in the scenario and general processes, "use cases," that are depicted in a set of relationships. These models give an intuitive view for the general user audience of the basic features of the proposed application and are a good way to start analysis of the elements of an application. Quality dimensions for UML Models include the following:

7.5.2.1 Conceptual completeness requires identification of all model components as indicated by the specific UML technique;

7.5.2.2 Conceptual correctness requires that all UML model components accurately reflect the process or object being represented;

7.5.2.3 Syntactic completeness requires all components are named and defined; and,

7.5.2.4 Syntactic correctness requires that all model components are treated consistent with the specific UML technique.

TABLE 6 Quality Dimensions and Examples for Health Informatics Object Models

Object Model Quality Dimension	Model Quality Characteristics
Enterprise awareness	The object model depicts the entire enterprise or object, or a component that can be seamlessly integrated with other models across the entire organization or component.
Conceptual correctness	The object model accurately reflects health or business concepts.
Conceptual completeness	The object model contains sufficient objects to describe the full scope of the target health or business domain.
Syntactic correctness	The objects displayed in the object model do not violate syntactic rules of the modeling language.
Syntactic completeness	All health or business object concepts are captured at appropriate points in the modeling process.

8. Practical Considerations in Modeling

8.1 The use of modeling to solve real-world problems should be approached as the strict use of methodology tempered by practical considerations.

8.2 *Understanding Feature Interaction and Other Anomalies*:

8.2.1 *Feature Interaction*—The phenomenon of feature interaction plagues the engineering of interoperable information architectures. The term feature interaction refers to those semantic inconsistencies in encapsulated concepts that are unrecognized at the time of assembly of components into a system. This is not a trivial problem. Feature interaction manifests as aberrant behavior during system testing or operation. Models reveal these semantic errors and changing the component models to reflect consistent behavior.

8.2.2 *Model Perspectives*—Models are only representations. Viewpoints and perspectives of that reality differ. While an activity model of clinic operation will appear differently depending upon whether it was constructed from a clinician's, administrator's, or patient's viewpoint, if prepared correctly all three models are accurate representations of that reality. Likewise all three models would be valuable contributions to the design of an information system intended to support that clinic operation. Models should be expected to have differing appearance and meaning depending upon the perspective from which these are derived.

8.3 *Modeling in Understanding Infrastructure Requirements*:

8.3.1 An understanding of the information technical architecture typically is developed early in the systems life cycle. Modeling facilitates this understanding by enabling the representation of large and complex concepts in a form more easily understood. An internationally accepted body of best practices in software engineering highlights the roles of models in the systems life cycle.

8.3.2 A variety of methods, techniques, and tools may be employed to produce clear, comprehensive, and explicit understanding of the business and technology environment. These models must accommodate the evolution of the enterprise's operations and facilitate integration of the information architecture with the changing business environment. The models are applicable to and usable by all constituencies within the organization.

8.4 *Modeling in Healthcare Reengineering and Business Process Improvement*:

8.4.1 Implicit in the term reengineering, is the intention to actively reconstruct the enterprise business processes in a new form. Examination and analysis of the business of healthcare means representing both the care delivery and the care management dimensions of the healthcare information domain. This representation must be performed in an integrated fashion using consistent notations that depict both concepts and relationships so that both dimensions can be clearly understood. The considerations provided elsewhere in this document involve the selection of methods, tools, and techniques that enable an efficient and effective representation of the enterprise. This representation is the starting point for the transition from conceptual understanding to implementing an information architecture designed to serve the future enterprise business functions.

8.5 *Modeling in Healthcare Data Representation*—In an enterprise, models present a relatively static view of the structures and relationships among those concepts comprising the enterprise. Process, activity, and data models typically are employed and present trade-offs in the way that concepts are handled. The data model depicts the structural representation, for example, vocabulary, images, graphics, etc., as well as the relationships among those represented concepts. The data model, particularly in large and complex enterprises or functions, follows the understanding built by modeling the enterprise or function processes and activities.

8.6 *Modeling in Healthcare Security, Privacy, and Confidentiality*—Modeling in the realm of privacy/confidentiality and security involves employing various techniques to identify those enterprise scenarios, activities, and data where compromises of confidentiality and security may occur and negatively impact the enterprise, its customers, or the population at large. Modeling contributes to effective risk management and the reduction or elimination of those potentially compromising situations. The model developer should build functionality and utility into the model and only then constrain the application for security in implementation.

8.7 *Modeling in Application Development*:

8.7.1 Modeling is a necessary activity in the development of all but the most trivial of applications. The health care domain is so large and complex that modeling is essential to the effective and efficient development of applications, modification and adaptation of software, and in systems integration. The framework for the application, as discussed in 5.5, needs careful definition with respect to the stated enterprise boundaries. This definition will inventory the range of concepts that will be needed in the application area and the common conventions that will need to be considered and modeled.

8.7.2 A very helpful way to organize this information is given in IEEE 1362-98 Concept of Operations, which should precede other life cycle documents that attend the various processes that contribute to the application focal area. While modeling is a major contributor to the requirements engineering aspects in both development and maintenance/evolution of an application, it has utility in the design and construction activities of the life cycle. Because an application may deal with and ancillary service area, it clearly must interoperate with such core areas of healthcare as the EHR, which contains the recorded observation about care that individuals have received; thus, evolution of a focal area, including resource management, must always keep that need to be consistent with core areas in mind in order to avoid the feature interaction effect noted in 8.2.1.

ANNEXES

(Mandatory Information)

A1. ORGANIZATIONAL CHECKLIST FOR MODEL ACTIVITIES

A1.1 The checklist shown in Fig. A1.1 determines a numerical score for the maturity of modeling within an

Organization Structural Requirements:

Quality Requirement	Explanation	Check if Present
Organizational commitment	Organization policy is to use modeling routinely for problem solving and system development.	
Stakeholder	A senior member of the organization is sponsoring the modeling effort.	
Model-master	One individual is accountable for the organizational modeling activities and responsible for models in the organization.	
Trained modelers	The organization has selected, trained and maintains competence of an appropriate number of modelers.	
Executive buy-in	Organization executives and senior managers understand the need for and benefits of modeling to solve organizational problems and create information systems.	

Modeling Process Requirements:

Quality Requirement	Explanation	Check if Present
Internal standards	The organization has adopted modeling policies and internal standard procedures for building, maintaining and using models.	
Toolsets	The organization has objectively selected modeling toolsets based on intended use and independent of personal preference for a specific methodology or technique.	
Version Control	The organization has established policies and procedures for managing model edits and revisions.	
Environment	The organization has facility resources (location, equipment, etc.) dedicated for modeling activities.	

Modeling Outcome Requirements:

Quality Requirement	Explanation	Check if Present
Complete	The modeling efforts accurately depict the necessary features of the objects being represented.	
Focus	The model efforts omit unnecessary or confounding detail.	
Understandable	The models are readily understandable by readers having a minimum of expertise in the modeling method or technique.	
Compliance	The models conform to standards or best practices for the particular modeling method or technique.	
Documentation	The models are accompanied by sufficient documentation, prepared in a clear and concise manner, to facilitate understanding.	
Usefullness	The models are used and contribute to improved business practices, products or services.	

TOTAL CHECKMARKS	
DETERMINE SCORE (DIVIDE TOTAL BY 15)	

FIG. A1.1 Organizational Checklist for Modeling Activities

organization.[7]

[7] After Donnabedian's structure, process, and outcome quality constructs.

A2. MODEL QUALITY CHECKLIST

A2.1 The checklist shown in Fig. A2.1 is a guide for determining the model quality.

A2.2 In addition to the general modeling quality checklist, Fig. A2.2 shows a checklist that is a guide for determining the quality of a data model.

	Model Review Item	Check if YES
1	Is there an organizational commitment to using the model for workflow, software, or database design?	
2	Have domain or subject matter experts been involved in developing the model?	
3	Is a procedure in place to deal with problems in the model and modeling process?	
4	Has the model been selected by comparison to other models?	
5	Does the model accommodate legacy systems?	
6	Does the model accommodate historical content?	
7	Does the model conform to standards for modeling method syntax and structure?	
8	Are the author/editor, title, version and date clearly stated on the model?	
9	Does the model conform to documentation standards?	
10	Does the model conform to presentation standards?	
11	Does the model fully cover the scope of the health or business domain for which it is intended?	
12	Is the model scope and viewpoint clearly stated?	
13	Is the model well-presented and understandable?	
14	Have all redundant features been removed from the model?	

Scoring: Count all checks and divide by 14. Report as a decimal score.

FIG. A2.1 General Model Quality Checklist

	Data Model Review Item	Check if YES
1	Was the model derived from a detailed functional model?	
2	Is a data management strategy in place in the organization and does the data model conform to this data management strategy?	
3	Are data views identified and understandable?	
5	Are subject areas identified and rational?	
At the Entity Level:		
1	Do all entities have meaningful and unambiguous names?	
2	Do all entities have clear and concise definitions?	
3	Is there an appropriate level of generalization of the entities?	
4	Is there an appropriate level of generalization of the relationships?	
5	Have relationships been correctly documented?	
6	Are all reference entities (for look-up tables) identified?	
7	Have all relationships that are derivable from data been removed?	
At the Key-Based Level:		
1	Have all many-to-many relationships be removed or resolved?	
2	Do all entities have complete primary keys?	
3	Are the number of keys for each entity manageable?	
4	Are all foreign keys identified?	
5	Are all primary keys appropriate to the entity?	
6	Are all primary keys unique in name, definition and meaning?	
7	Are all primary keys minimal?	
8	Are all primary keys stable?	
9	Are surrogate keys used appropriately?	
At the Fully Attributed Level:		
1	Are all needed attributes present?	
2	Are unnecessary attributes omitted?	
3	Do all attributes have meaningful and unambiguous names?	
4	Do all attribute names conform to organizational documentation standards?	
5	Do all attributes have clear and concise definitions?	
6	Are alternate keys and inversion entries employed where needed to enhance usability?	
7	Are all attributes characterized correctly (e.g. as date elements)?	
8	Are typical values for attributes noted where required for understanding?	
For transactional logical data models:		
1	For transactional databases, are all logical model entities and physical model tables fully normalized?	
For analytical logical data models:		
1	For analytical databases, has the structure (e.g. star, starflake, or snowflake schema) been correctly represented?	
For transformational data models:		
1	Has the appropriate target database management system been specified?	
2	Has the model been appropriately optimized (e.g. through denormalization) for performance on the target database management system?	

FIG. A2.2 Data Model Quality Checklist

A3. PROCESS MODEL QUALITY CHECKLIST

A3.1 Data flow diagramming is a modeling method that is used to document the movement and processing of information within a business or organization. Often a series of data flow diagrams (DFDs) describe the current or future operation of an organization's data processing system and provides a blueprint for systems to be developed. The data flow diagram describes the following:

A3.1.1 The data processing functions, for example, Fig. A3.1, Input Patient Data,

A3.1.2 The data used or created by the data processing system, for example, Reference Values, and

A3.1.3 The objects, persons, or departments that interact with the data, for example, CPCRG customer, and

A3.1.4 The flow of data among the functions, data stores, and sources/destinations of the data.

A3.2 While all objects must have a unique and descriptive name, multiple copies of some components may be used within a diagram and at different levels of a model (as in a decomposed process) to promote clarity. Fifteen rules describe the correctness and facilitate the understanding of data flow diagrams shown in Fig. A3.2.

A3.3 In addition to the general modeling quality checklist, the checklist shown in Fig. A3.2 is a guide for determining the quality of a process (data flow diagram) model.

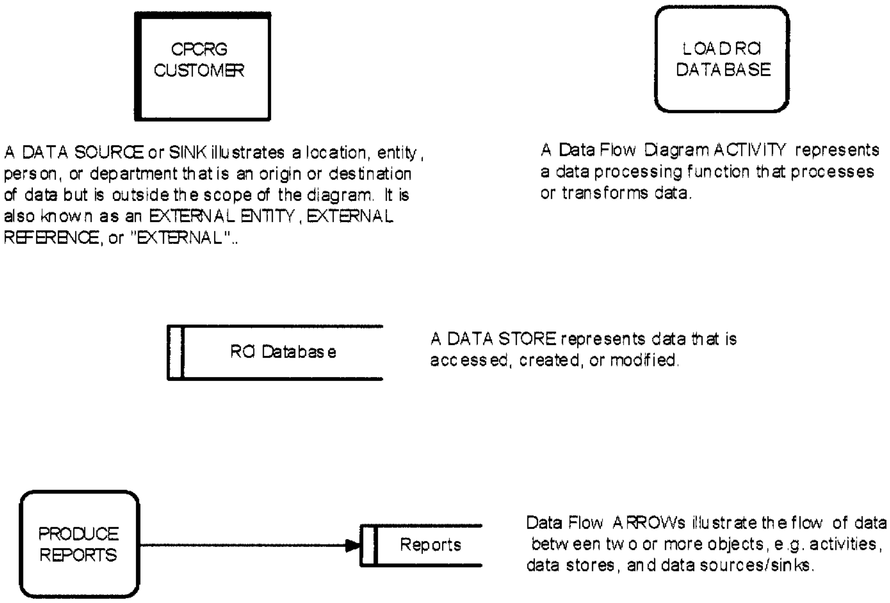

FIG. A3.1 Four Graphic Elements Used in Gane and Sarson Data Flow Diagramming

Process (DFD) Model Review Item		Check if YES
1	Is the activity name a Verb Phrase that describes the tasks being performed?	
2	Are the inputs to an activity are always different from the outputs?	
3	There are no activities that have only inputs or only outputs.	
4	No data moves directly from one data store to another without going through a process.	
5	No data moves directly from a data source to a data store without going through a process.	
6	No data moves directly from a data source to a data sink without going through a process.	
7	The data store, source and sink names are noun phrase that accurately describes the contents.	
8	All data flows are one-directional (no bidirectional arrows).	
9	A forked or split data flow delivers exactly the same data to two or more different destinations.	
10	A joined data flow receives exactly the same data from two or more different sources and passes these data to a common destination.	
11	Aggregated data flows are appropriately named.	
12	No data flow returns to the originating process without going through one or more additional processes.	
13	All data flows to a data store indicate that a create, update, or edit process occurs.	
14	All data flows from a data store indicate a retrieve, read, or use process occurs.	
15	When named, all data flows are named with a noun phrase label.	

Scoring: All boxes must be checked.

FIG. A3.2 Process Model Quality Checklist Using a Data Flow Diagram Example

REFERENCES

(**1**) Donabedian, A., *Explorations in Quality Assessment and Monitoring*, Vol I, "The definition of quality and approaches to its assessment," Ann Arbor:Health Administration Press, 1980.

(**2**) Reingruber, M. C. and Gregory, W. W., "The Data Modeling Handbook," New York:Wiley, 1994.

(**3**) Zachman, J. A., Framework for Information Systems Architecture, *IBM Systems Journal 26/3*, 1987.

(**4**) Simsion, Graeme, "Data Modeling Essentials," New York:Thomson, 1994.

(**5**) Martin, J., "Strategic Information Planning Methodologies," 2nd edition, Englewood Cliffs:Prentice Hall, 1989.

(**6**) Martin, J., "Diagramming Techniques for Analysts and Programmers," Englewood Cliffs:Prentice Hall, 1985.

(**7**) Booch, G., Rumbaugh, J., and Hacobson, I., "The Unified Modeling Language User Guide," Reading, MA:Addison-Wesley, 1999.

(**8**) Bruce, T. A., "Designing Quality Databases with IDEF1X Information Models," New York:Dorset House, 1992.

(**9**) Jacobson, I., Booch, G., and Rumbaugh, J., "The Unified Software Development Process," Reading, MA:Addison-Wesley, 1999.

(**10**) Pressman, R. S., "A Manager's Guide to Software Engineering," New York: McGraw-Hill, 1993.

(**11**) Strassmann, P. A., "The Business Value of Computers: An Executive's Guide," New Canaan, CT:Information Economics Press, 1990.

(**12**) Bruce, P., and Pederson, S. M., "The Software Development Project," New York:Wiley, 1982.

(**13**) Metzger, Philip W. with John Boddie (contributor), "Managing a Programming Project: People and Processes," Englewood Cliffs: Prentice-Hall ECS Professional, 1985.

(**14**) Jones, T. C., "Estimating Software Costs," New York:McGraw-Hill, 1998.

(**15**) Bourke, M. K., "Strategy and Architecture of Health Care Information Systems," New York: Springer, 1994.

(**16**) Integration Definitions for Functional Modeling (IDEF0) Federal Information Processing Standard 183, Gaithersburg, MD: National Institute of Standards and Technology, 1993.

(**17**) International Standard ISO 8601 Data elements and interchange formats—Information interchange—Representation of dates and times. First edition 1998-06-15. Ref No ISO 8601: 1988(E).

(**18**) DeMarco, T., "Structured Analysis and Systems Specification," Englewood Cliffs:Yourdon, Inc., 1978.

(**19**) Hoffer, J., George, J., and Valacich, J., "Modern Systems Analysis and Design," 2nd ed. Reading:Addison-Wesley, 1998.

(**20**) Bourke, M. K., "Strategy and Architecture of Health Care Information Systems," New York:Springer, 1994.

(**21**) Bachman, C., "Data Structure Diagrams," Database 1/2: pp.4-10, 1969.

(**22**) Bachman, C., "The Evolution of Storage Structures," Communications of the ACM 15/7: pp. 628-634, July 1972.

(**23**) Chen, P., "The Entity-Relationship Model—Toward a Unified View of Data," ACM Transactions, Database Systems 1/1:pp. 9-36, Mar 1976.

(**24**) Fleming, C., and VanHalle, B., "Handbook of Relational Database Design.," NY:Addison-Wesley, 1989.

(**25**) Silvershatz, A., Forth, H., and Sudarshan, S., "Database System Concepts," 3rd ed. New York:WCB McGraw-Hill, 1996.

(**26**) Silverston, L., Inmon, W., and Graziano, K., "The Data Model Resource Book," New York:Wiley, 1997.